Geostat Summer School

QGIS

An Open Source Desktop GIS

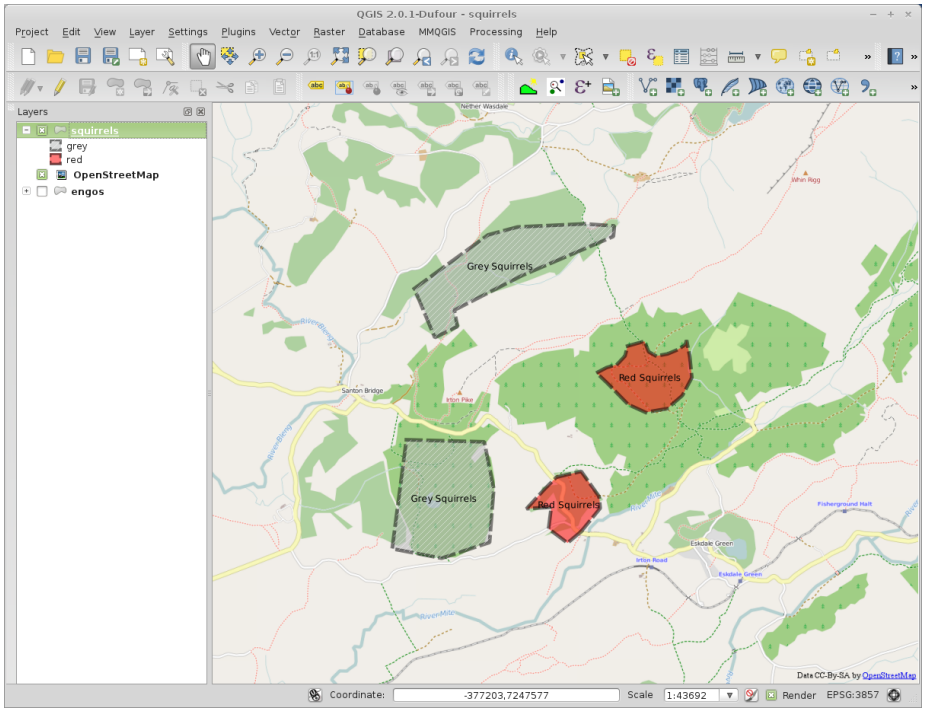
Intro

Features

- Cross-platform Lin/Win/Mac Desktop GIS
- Works with Geospatial Standards
- Written in C++
- Embedded Python

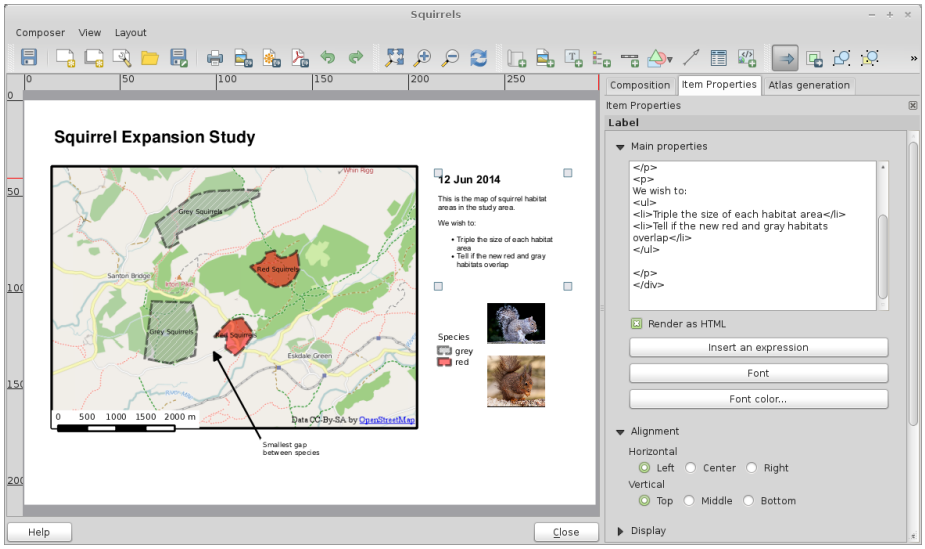
Screenshot

Typical screen



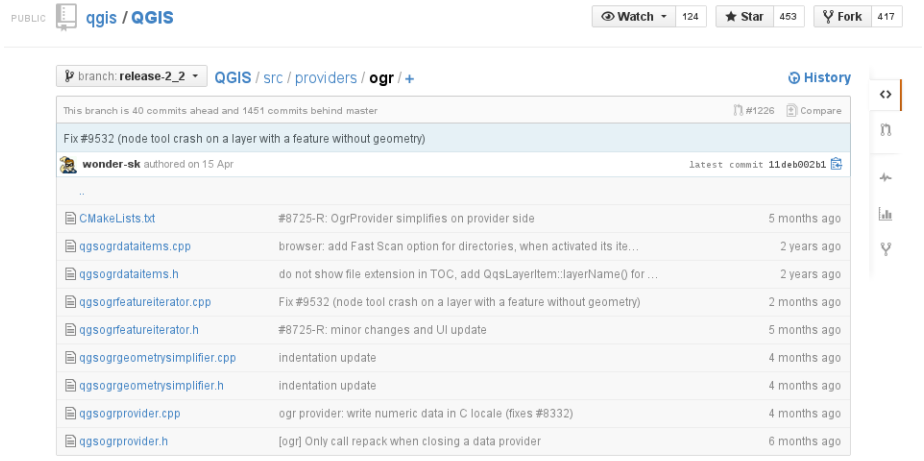
Composer

Map Composer



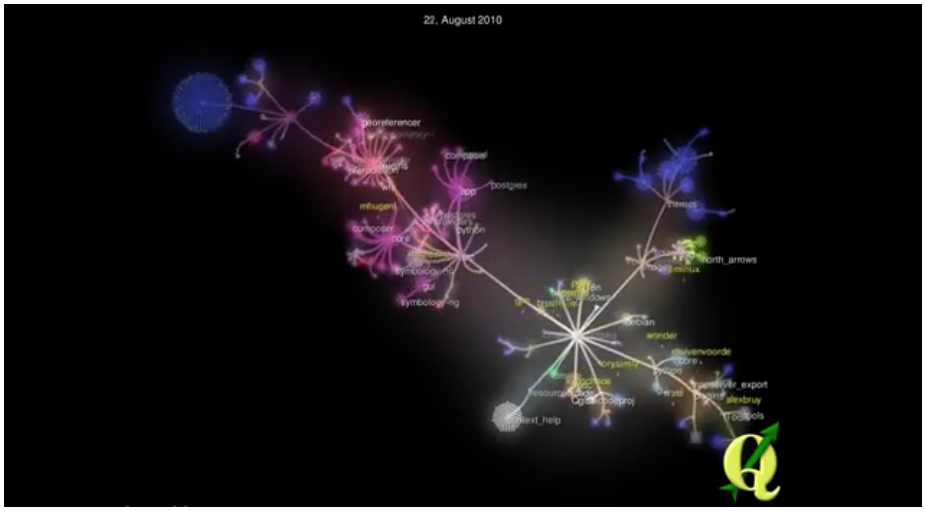
C++

C++



Development

Visualisation



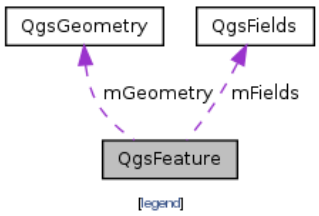
Feature

Structure

The feature class encapsulates a single feature including its id, geometry and a list of field/values attributes. More...

```
#include <qgsfeature.h>
```

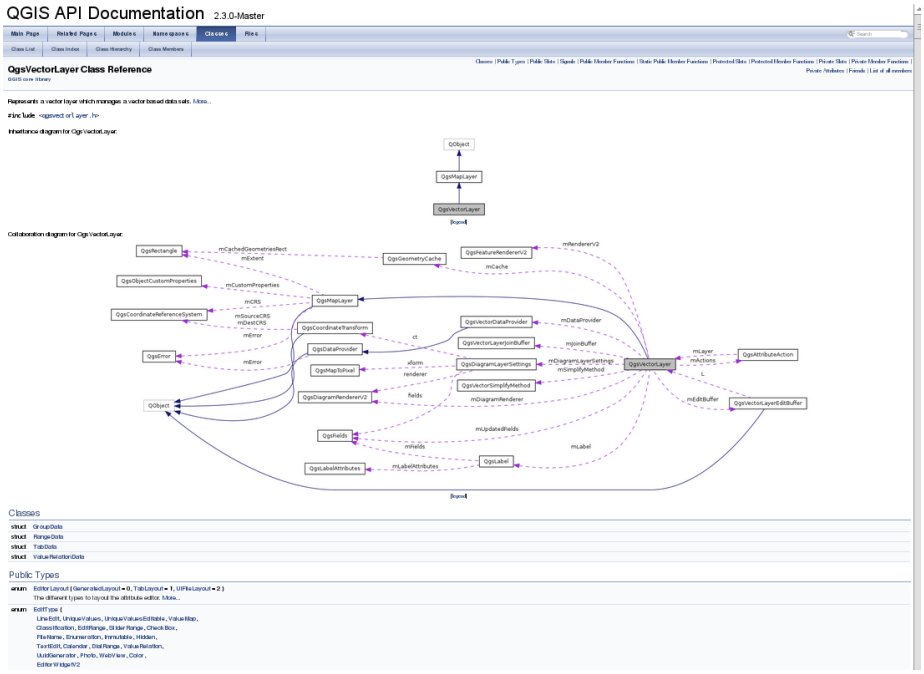
Collaboration diagram for QgsFeature:



[legend]

Vector Layer

Structure



This is the structure of a vector layer

Feature methods

Method list

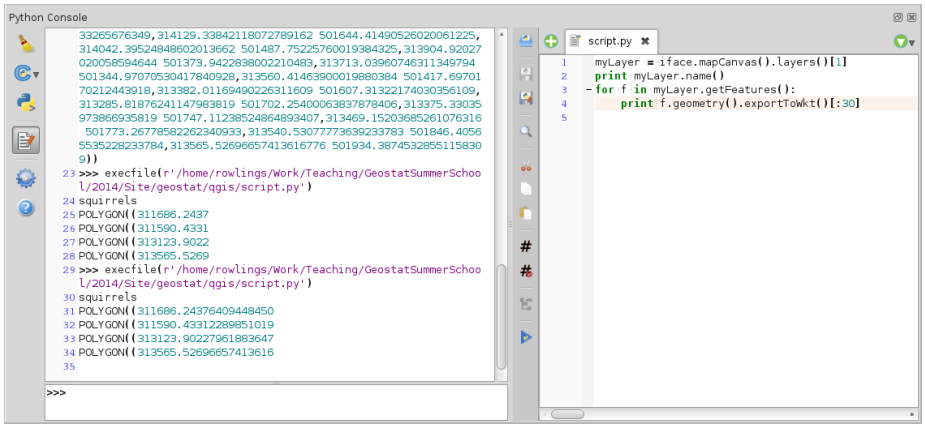
	Set the validity of the feature. More...
QgsGeometry *	**geometry** () const Get the geometry object associated with this feature. More...
QgsGeometry *	**geometryAndOwnership** () Get the geometry object associated with this feature The caller assumes responsibility for the QgsGeometry*'s destruction. More...
void	**setGeometry** (const **QgsGeometry** &geom) Set this feature's geometry from another **QgsGeometry** object (deep copy) More...
void	**setGeometry** (**QgsGeometry** *geom) Set this feature's geometry (takes geometry ownership) More...
void	**setGeometryAndOwnership** (unsigned char *geom, size_t length) Set this feature's geometry from WKB. More...
void	**setFields** (const **QgsFields** *fields, bool **initAttributes**=false) Assign a field map with the feature to allow attribute access by attribute name. More...
const **QgsFields** *	**fields** () const Get associated field map. More...
bool	**setAttribute** (const QString &name, QVariant value) Insert a value into attribute. More...

Python Console

Console

Python Editor

Console with Editor



Squirrels

Red and Grey

CC-licensed from flickr users imh and hisgett

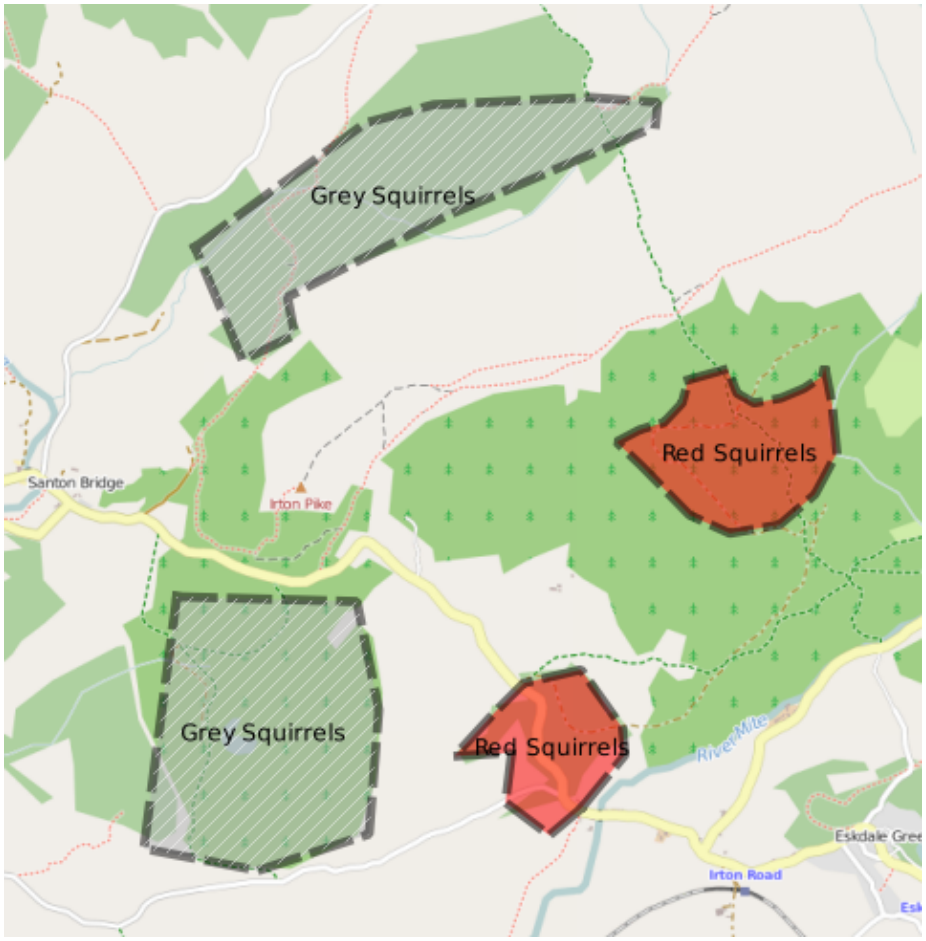
Squirrels

Red and Grey

CC-licensed from flickr users king-edward and hisgett

Status

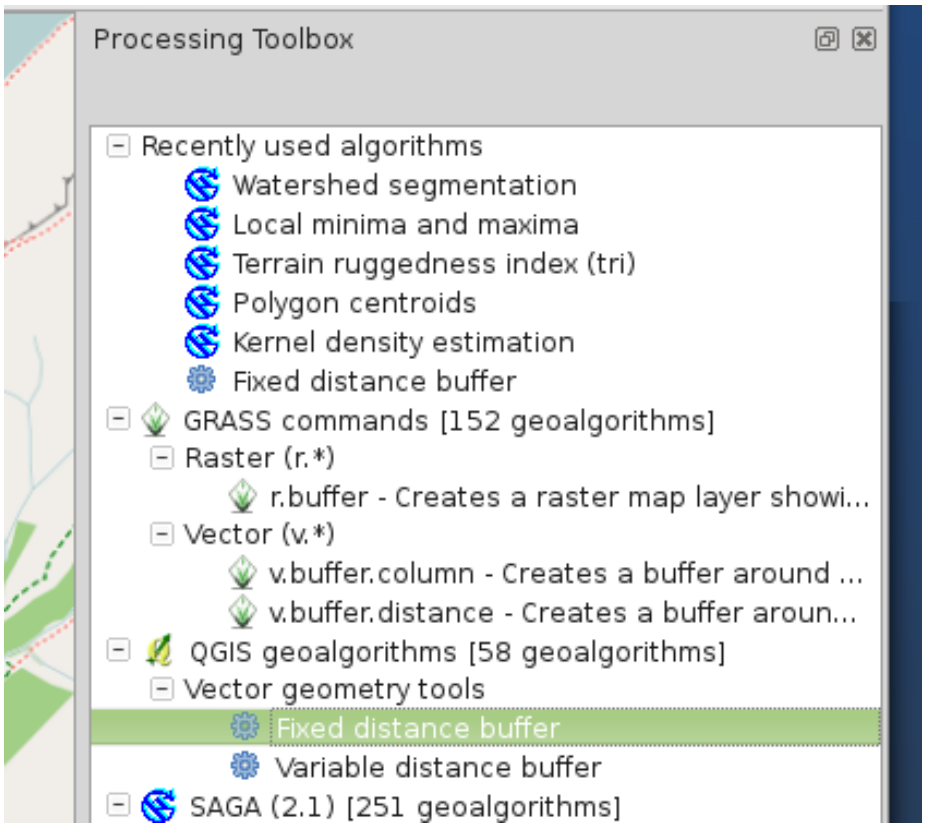
Distribution



- What if populations triple and areas triple?
- Can we buffer the regions and check for overlap?

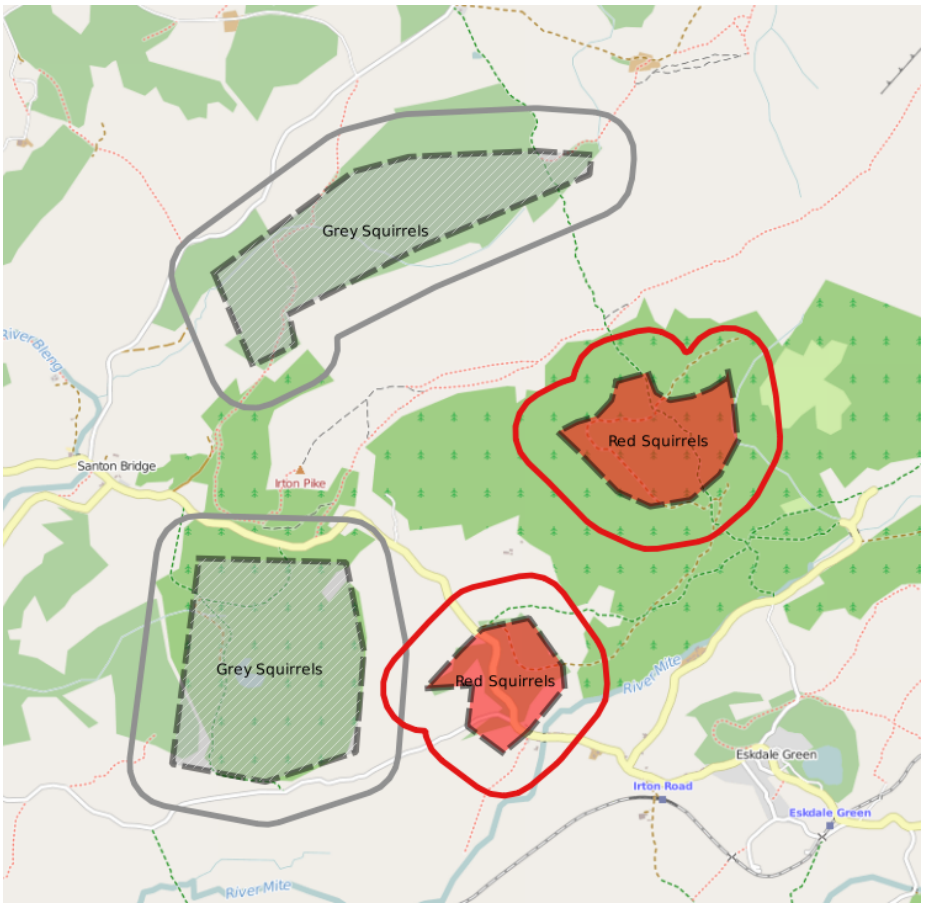
Buffering

Options



Fixed Buffer

Styled. . .



- How do we find a buffer width that scales an area?

Strategy

Binary Search

- Area of buffer increases as width increases
- Use a binary search algorithm to find w such that area of buffered region = S * area of region

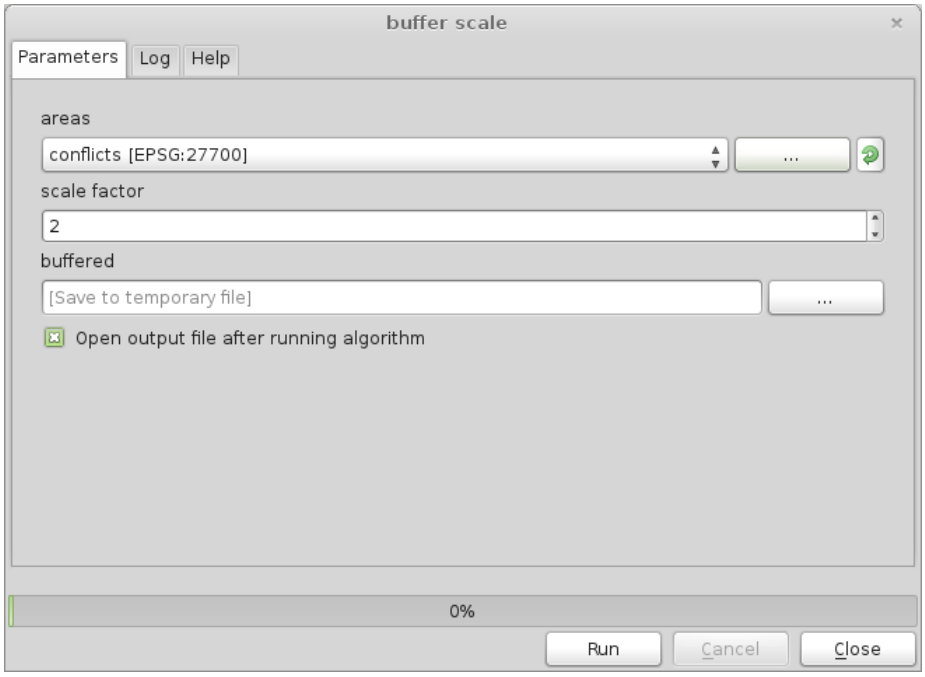
QGIS Processing Scripts

Easy

```
##[BSR scripts]=group
##areas=vector
##scale_factor=number 2
##buffered=output vector

from PyQt4.QtCore import *
from qgis.core import *

import processing
from processing.core.VectorWriter import VectorWriter
from bsrutils import rescaleBuffer

areas_layer = processing.getobject(areas)
areas_features = processing.getfeatures(areas_layer)
fields = areas_layer.pendingFields().toList()
writer = VectorWriter(buffered, None, fields, POLYGON, areas_layer.crs() )

for feature in areas_features:
  outFeat = rescaleBuffer(feature, scale_factor)
  writer.addFeature(outFeat)
```

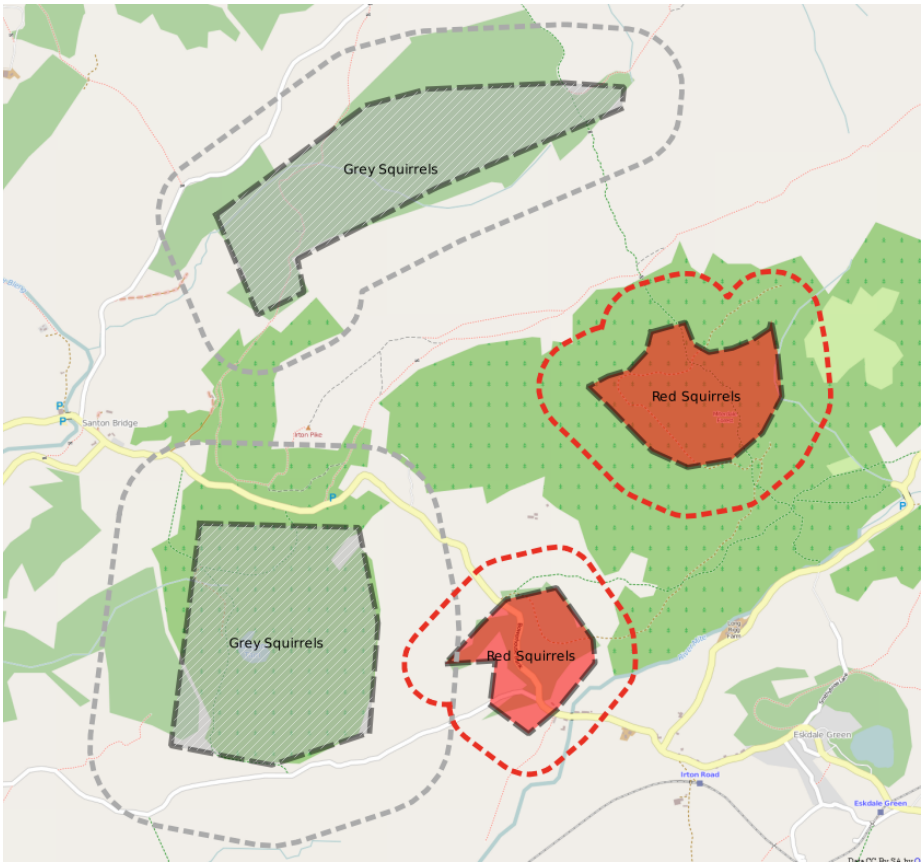
ui

Gets this


buffer scale
Parameters
Log
Help
areas
conflicts [EPSG:27700]
...
scale factor
2
buffered
[Save to temporary file]
...
Open output file after running algorithm
0%
Run
Cancel
Close

Run

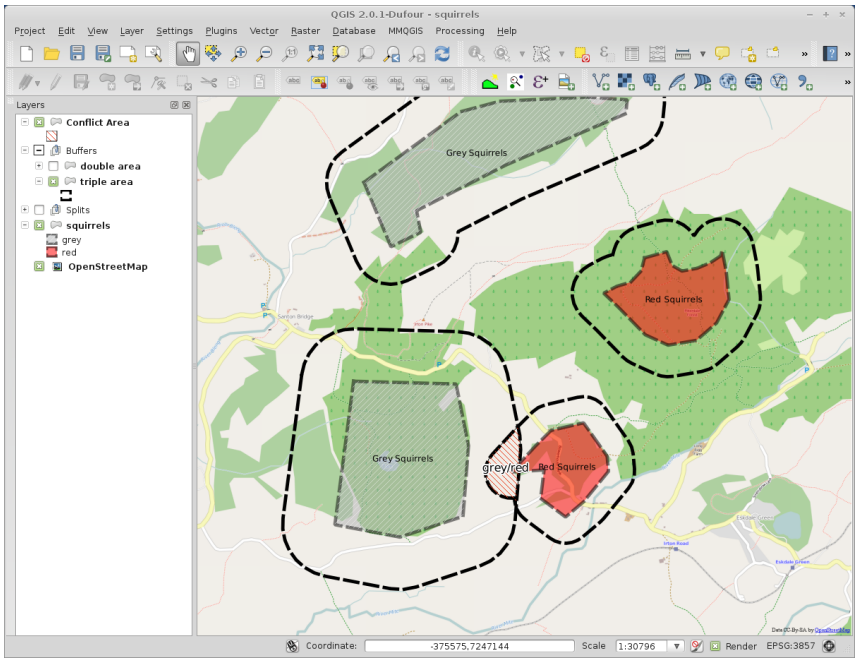
and get



Conflict Area

Getting the conflict area

- Split layer on attribute value
- Intersection of those layers



- Can we automate this?

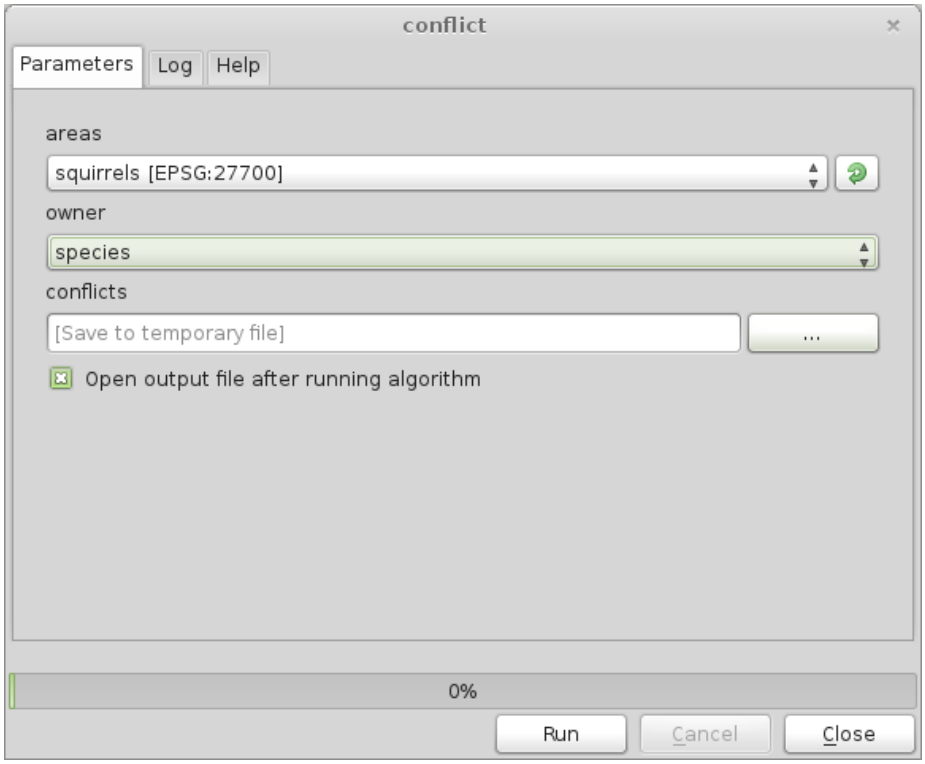
Conflict processing script

outline

```
##[BSR scripts]=group
##areas=vector
##owner=field areas
##conflicts=output vector
...
for aFeature in features:
  for bFeature in features:
     if ownerValue(aFeature) != ownerValue(bFeature):
        if aFeature.geometry().intersects(bFeature.geometry()):
            outFeature.setGeometry(aFeature.geometry().intersection(bFeature.geometry()))
            writer.addFeature(outFeature)
```

Conflict UI

Dialog



Two steps

Can we do better?

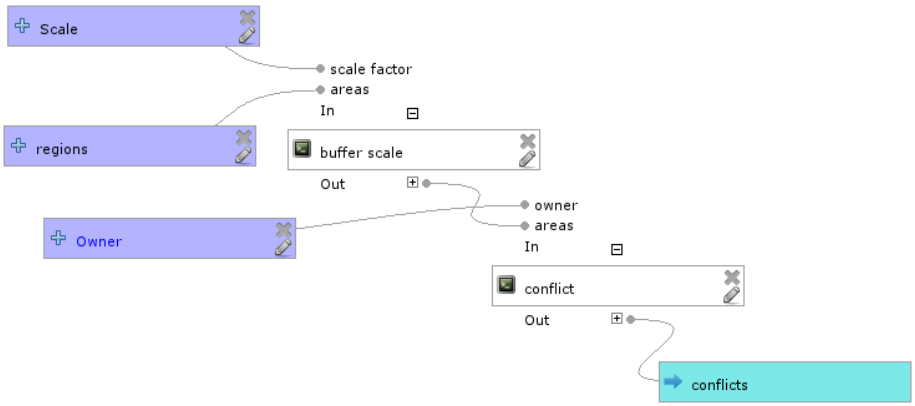
- Buffer areas by scale factor
- Compute conflict areas

Models

- Allows connected algorithms
- Can use any algorithms in the processing toolbox
- Graphical tool connecting inputs, algorithms, outputs

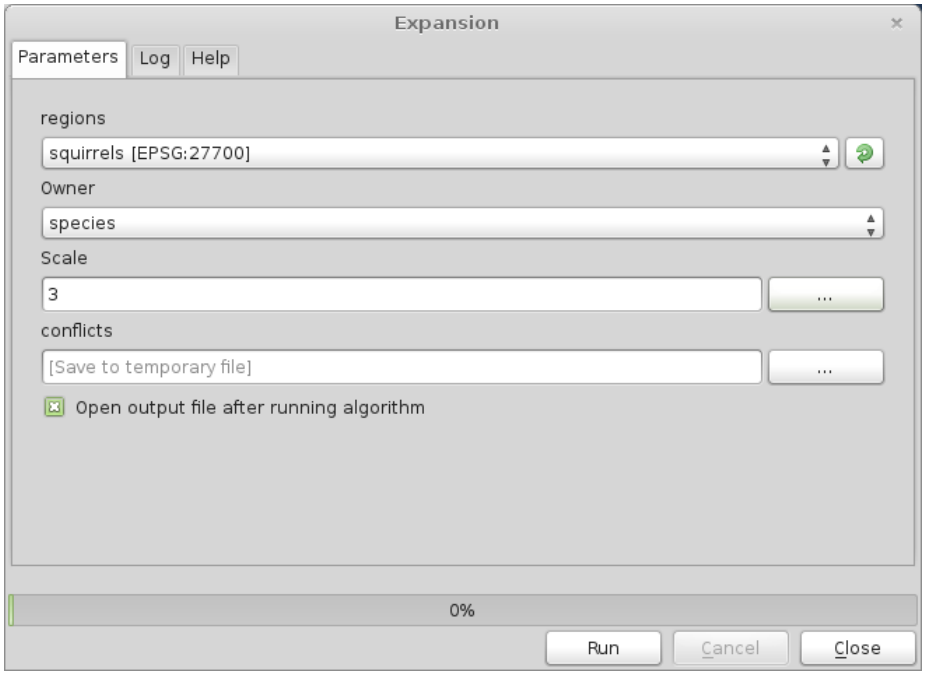
Model builder

Expansion Model



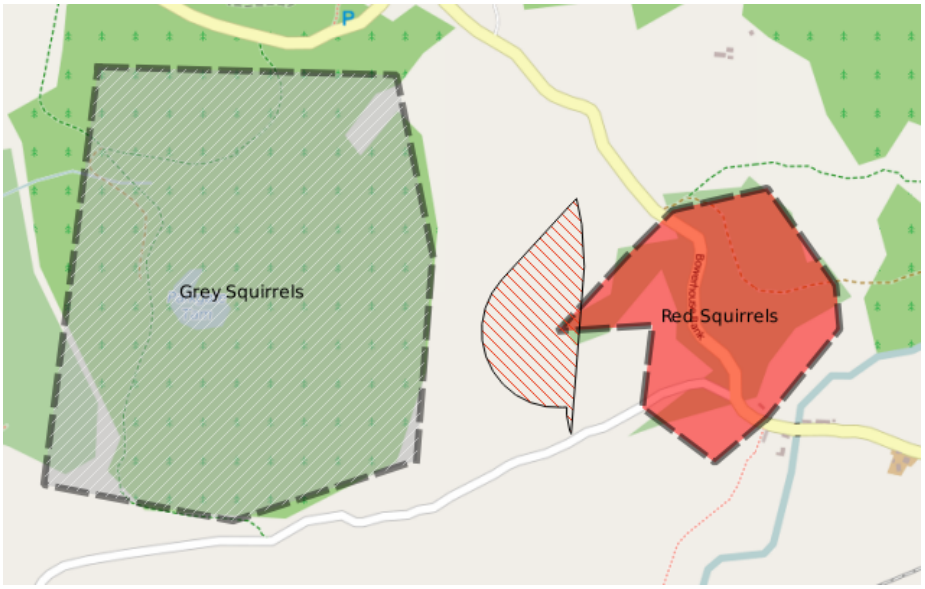
Model builder

Expansion UI



Conflict Output

One step, one extra layer



Other advanced QGIS features

Good stuff

Python Plugins

- Build Custom GUIs
- Create Menus
- Define New Layer Types
- Define New Renderers

R Processing Scripts

- Leverage R-spatial
- Examples Included

Feature Editing/Cleanup

Spatial Database Integration

OGC Services